

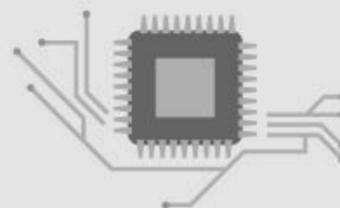
タグ情報を使ってリスクエリアを可視化し、 効率的にバグを検出するテスト手法

ソニーデジタルネットワークアプリケーションズ株式会社
事業推進部 品質推進課

福西 章記



<https://www.sonydna.com>



はじめに

本資料について

- 本発表は、『可視化したリスクエリアに対して集中的/効率的にテストする』ことで、バグの検出力を大きく上げることが出来た事例を紹介する内容となります。

会社概要



事業内容

ソフトウェア技術の追求と
新規領域への挑戦

ソニーデジタルネットワークアプリケーションズ株式会社 (SDNA)

所在地	神奈川県横浜市西区みなとみらい5丁目1番1号	資本金	1億円	ソニー株式会社	100%出資
事業内容	ソフトウェアの企画・開発・商品化	年商	69億円	(2024年度実績)	
設立	2000年8月1日	社員数	309名	(2025年4月1日現在)	

特有の用語について

- 本資料では以下の特有の用語を使用します。

No	用語	内容
1	タグ	機能仕様やテスト内容/リスク観点を連想できる用語をワード化したもの。
2	タグマトリクス	タグの組合せの出現カウント数をマトリクス化したもの。スコアの高いところをリスクエリアとして扱う。
3	MW	ミドルウェアモジュールの略。製品のUI面とHW面の中間の制御を行う。主に「ファイルの書き出し/読み込み/DB管理」の3つの機能を持つ。
4	PMG	パワーマネジメントモジュールの略。製品の電源管理の制御を行う。
5	Network	ネットワークモジュールの略。製品のネットワーク関連機能の制御を行う。
6	UI面	ユーザーインターフェースモジュールの略。製品の各画面や各ボタンなど機能に関する全般を取り扱っている。
7	テスト設計	テストケース作成からシート作成までの工程のこと。
8	評価実施	テスト実施のこと。

目次

1. バグ検出における課題の発生
2. 改善施策の検討
3. タグ情報を使ったテスト手法の導入と成果
4. 成果の横展開
5. 今後に向けた改善活動

1. バグ検出における課題の発生

活動開始の経緯/発端

- MWテストとしてバグ検出力の課題を確認
- 専門チームなのに全体の7.1%という低い状況



活動開始の経緯/発端

- MWテストとしてバグ検出力の課題を確認
- 専門チームなのに全体の7.1%という低い状況

MW修正バグ数：
892件
(*FY16以降PJ
が対象)

バグ検出力向上を目的と
した活動が始まった。

MWチームバグ検出数：
63件 ※全体の7.1%

2. 改善施策の検討

バグ検出力向上を実現するために

1. まずはバグの分析を実施

- 1ヶ月以上かけて、約1000件のMW開発チームが修正したバグレポート内容を地道に確認した



バグ検出力向上を実現するために

1. まずはバグの分析を実施

- 1ヶ月以上かけて、約1000件のMW開発チームが修正したバグレポート内容を地道に確認した

2. 機能や手順の傾向を発見

- MWモジュールの特性である「書き出し/読み込み/ファイル管理」の組合せが関連していた



バグ検出力向上を実現するために

1. まずはバグの分析を実施

➤ 1ヶ月以上かけて、約1000件のMMV開発チームが修正

■ 傾向の例)

ファイル書き出し処理中、カスタムキー割り込み、録画開始直後、コンテンツ編集、など。

⇒機能仕様やテスト内容/リスク観点を連想できる分類（タグ）にすることにした。

込み/

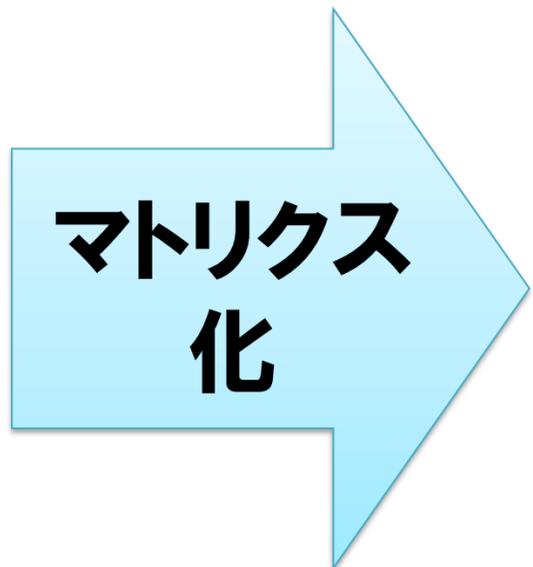


バグ検出力向上を実現するために

3. データのマトリクス化

- 1つのバグが複数の分類（タグ）に強く関連している事が多く、組合せ可視化のために2次元の表（マトリクス）とした

	スコア
タグ1	1 4
タグ2	1 2
タグ3	1 2
タグ4	1 6



	タグ1	タグ2	タグ3	タグ4
タグ1		1	3	1 0
タグ2			7	4
タグ3				2
タグ4				

バグが出やすいところが見えてきた (リスクエリア)



バグ検出力向上を実現するために

3. データのマトリクス化

- 1つのバグが複数の分類合せ可視化のために2次

この案を使ったテスト
プロセスを構築する事にした

	スコア
タグ1	1 4
タグ2	1 2
タグ3	1 2
タグ4	1 6



	タグ1	タグ2	タグ3	タグ4
タグ1		1	3	1 0
タグ2			7	4
タグ3				2
タグ4				



3. タグ情報を使ったテスト手法の導入と成果

バグが検出されている組み合わせを集計

■ 具体例) バグレポートにタグが付与された状態

No.	バグ内容	付与されたタグ
1	電源ON直後に高フレームレートで動画撮影すると撮影開始できない。	[起動処理中][電源OFF/ON][動画撮影] [特定フレームレート]
2	書出し処理中に検索再生(表示)すると表示が遅れる。	[連写][ファイル書出し処理中][検索再生]
3	書出し処理中に日付跨ぎの再生(表示)をすると表示されない。	[連写][ファイル書出し処理中][日付画面] [日付跨ぎ]
4	書出し処理中に検索再生画面で電源スイッチをOFFするとエラー表示されたフリーズする。	[連写][ファイル書出し処理中][検索再生] [電源OFF/ON]

条件が合うタグを、バグに付与する

バグが検出されている組み合わせを集計

■ データ集計方法

■ 以下の例の場合

★前頁のタグをマトリクス化

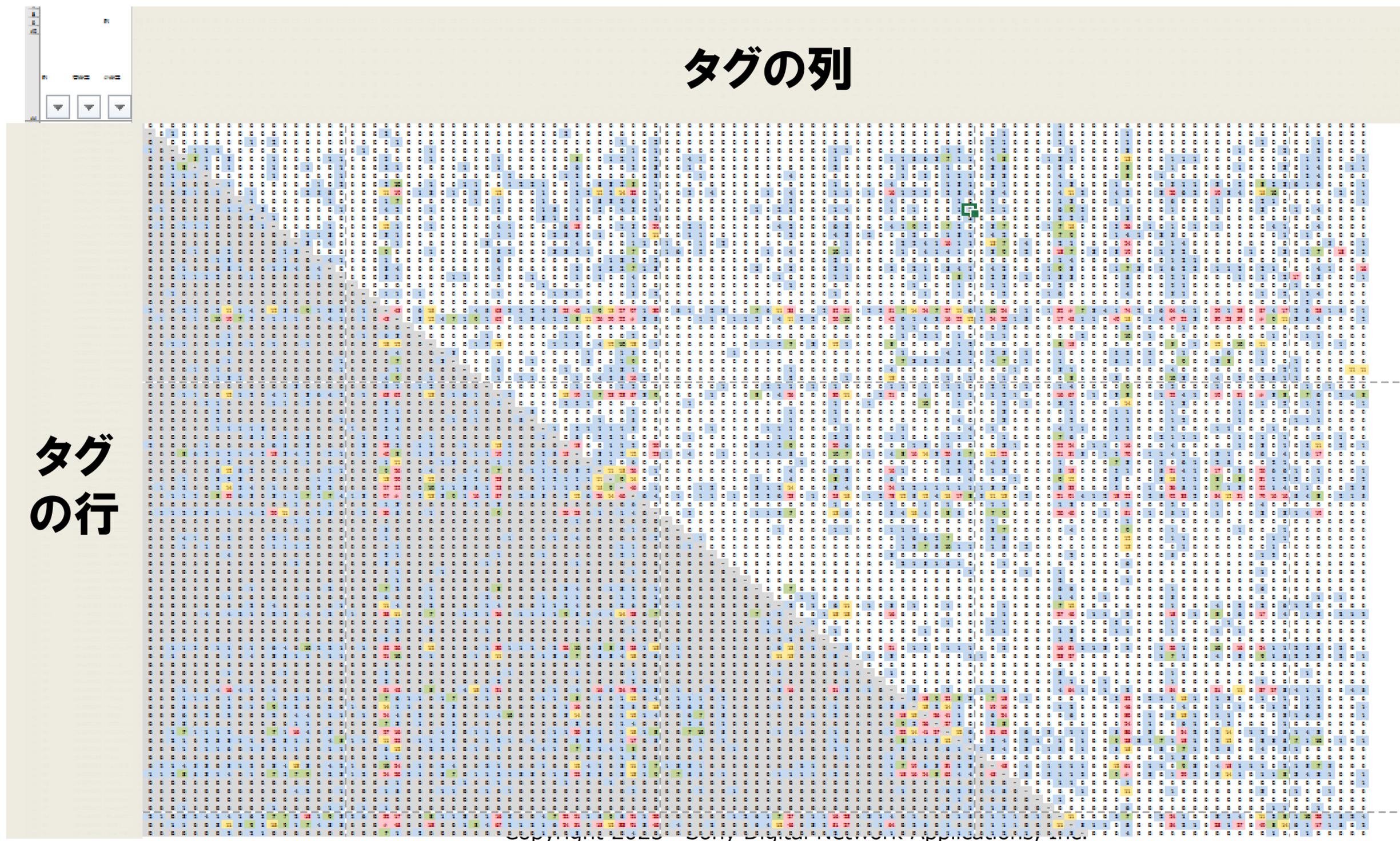
No.	付与されたタグ
1	[起動処理中] [電源OFF/ON] [動画撮影] [特定フレームレート]
2	[連写] [ファイル書出し処理中] [検索再生]
3	[連写] [ファイル書出し処理] [日付画面] [日付跨ぎ]
4	[連写] [ファイル書出し処理中] [検索再生] [電源OFF/ON]

	起動処理中	電源OFF/ON	動画撮影	特定フレームレート	連写	ファイル書出し処理中	検索再生	日付画面	日付跨ぎ
起動処理中		1	1	1					
電源OFF/ON									
動画撮影									
特定フレームレート									
連写									
ファイル書出し処理中									
検索再生									
日付画面									
日付跨ぎ									

集計する

バグが検出されている組み合わせを集計

- タグマトリクス実物：バグ1000件以上、タグ約100個で作成



- 【凡例】
- 白：バグの発生無し
 - 青：バグ5件以下
 - 緑：バグ10件以下
 - 黄：バグ15件以下
 - 赤：バグ16件以上

タグマトリクスによる仕組み導入の成果

- PJ で導入を開始したところ、バグ検出力で最大6倍、平均で3倍となった
 - 検出力 \Rightarrow 検出率 = バグ検出数 / テスト実施工数
 - 機能仕様テストチームの検出力0.010(*)を基準に算定
- (*)100.0hにつき1件のバグを検出するペース。
※機能仕様テストチームはバグが出る機能も出ない機能も網羅的にテストしている
- 最大実績：検出力 0.067 (約6倍)
 - 平均実績：検出力 0.032 (約3倍)

4. 成果の横展開

タグマトリクスによる手法の横展開

- MWのテストで成果が出たため、他のテストでも本手法を導入してほしいという要望をもらった。

タグマトリクスによる手法の横展開

➤ 「PMGテスト」 ⇒ 「Networkテスト」 ⇒ 「UI面テスト」
の順番に導入

テスト対象	主な取り組み	バグ検出率(導入年度の数値)
PMG	バグ傾向が有り、タグの生成が容易にできたので大きな課題も無く導入できた	0.030(約3倍) ~ 0.098(約10倍)
Network	バグ傾向は有るが、有識者を加えて知見不足を補う必要があった	0.025(約3倍) ~ 0.093(約9倍)
UI面	バグ傾向が無いため、『タグ作成基準』を定義して、様々な機能やリスクの観点に対応できるようにする事で導入できた	0.018(約2倍) ~ 0.103(約10倍)

※MW時と同様、機能仕様テストチームの検出力
0.010(*)を基準に算定

タグマトリクスによる手法の横展開

➤ 「PMGテスト」 ⇒ 「Networkテスト」 ⇒ 「UI面テスト」
の順番に導入

テスト対象	主な取り組み	バグ検出率(導入年度の数値)
PMG	バグ傾向が有り、タグの生成が容易にできたので大きな課題も無く導入できた	0.030(約3倍) ~ 0.098(約10倍)
Network	バグ傾向は有るが、有識者を加えて知見不足を補正した事で導入できた	0.025(約3倍) ~ 0.097(約9倍)
UI面	バグ傾向は有るが、機能仕様テストチームの検出力を向上させた事で導入できた	0.018(約2倍) ~ 0.103(約10倍)

他のテストでも同様の導入効果を得ることができている

※MW時と同様、機能仕様テストチームの検出力
0.010(*)を基準に算定

5. 今後に向けた改善活動

各プロセスのLLM化を推進

- **タグ付与とテスト設計**をAIに置き換えられるかを検討中
 1. タグ付与では、自動付与のメリットを活かして、より多くの分析をより短時間で行えるようにしていく
 2. テスト設計では、バグの狙いどころの精度を向上させて、より多くのバグを検出できるようにしていく

Appendix

タグ作成フロー

- 項目を分類できるようにするためにタグリストを作成
 - 以下フローで作成



- MW評価で必要そうなキーワードをとにかくリストアップ
 - ・テストの経験による思い付き
 - ・バグをベースにした抽出

⇒約 400 個洗い出し

- 文字の表記ブレや意味が同一のキーワードをマージ

⇒約 200 個に絞り込み

- MW全般で、リスク項目を分類できるようにするためのタグを作成
 - ・MWに関連しないタグの削除
 - ・ポリシーに関するタグの追加
 - ・タグの意味を考慮した統合と分割
 - ・開発者レビュー

⇒約 100 個にして完成

タグ作成フロー

■ 項目を分類できるようにするためにタグリストを作成

● タグの例)

- 起動処理中
- ファイル書き出し処理中
- カスタムキー割り込み
- 録画開始直後
- コンテンツ終端
- 日付跨ぎ
- 特定画サイズ
- コンテンツ編集
- 電源OFF/ON
- メディア挿抜

など 全体で約 1 0 0 個

**機能仕様やテスト内容/
リスク観点を連想できる
分類でワード化**

バグが検出されている組み合わせを集計(1/4)

■ データ集計方法

■ 以下の例の場合

★前頁のタグをマトリクス化

No.	付与されたタグ
1	[起動処理中][電源OFF/ON][動画撮影][特定フレームレート]
2	[連写][ファイル書出し処理中][検索再生]
3	[連写][ファイル書出し処理中][日付画面][日付跨ぎ]
4	[連写][ファイル書出し処理中][検索再生][電源OFF/ON]

	起動処理中	電源OFF/ON	動画撮影	特定フレームレート	連写	ファイル書出し処理中	検索再生	日付画面	日付跨ぎ
起動処理中		1	1	1					
電源OFF/ON									
動画撮影									
特定フレームレート									
連写									
ファイル書出し処理中									
検索再生									
日付画面									
日付跨ぎ									

集計する

バグが検出されている組み合わせを集計(2/4)

■ データ集計方法

■ 以下の例の場合

★前頁のタグをマトリクス化

No.	付与されたタグ
1	[起動処理中][電源OFF/ON][動画撮影] [特定フレームレート]
2	[連写][ファイル書出し処理中][検索再生]
3	[連写][ファイル書出し処理中][日付画面][日付跨ぎ]
4	[連写][ファイル書出し処理中][検索再生][電源OFF/ON]

	起動処理中	電源OFF/ON	動画撮影	特定フレームレート	連写	ファイル書出し処理中	検索再生	日付画面	日付跨ぎ
起動処理中		1	1	1					
電源OFF/ON			1	1					
動画撮影									
特定フレームレート									
連写									
ファイル書出し処理中									
検索再生									
日付画面									
日付跨ぎ									

集計する

バグが検出されている組み合わせを集計(3/4)

■ データ集計方法

■ 以下の例の場合

★前頁のタグをマトリクス化

No.	付与されたタグ
1	[起動処理中][電源OFF/ON][動画撮影] [特定フレームレート]
2	[連写][ファイル書出し処理中][検索再生]
3	[連写][ファイル書出し処理中][日付画面][日付跨ぎ]
4	[連写][ファイル書出し処理中][検索再生][電源OFF/ON]

	起動処理中	電源OFF/ON	動画撮影	特定フレームレート	連写	ファイル書出し処理中	検索再生	日付画面	日付跨ぎ
起動処理中		1	1	1					
電源OFF/ON			1	1					
動画撮影				1					
特定フレームレート									
連写									
ファイル書出し処理中									
検索再生									
日付画面									
日付跨ぎ									

集計する

バグが検出されている組み合わせを集計(4/4)

■ データ集計方法

■ 以下の例の場合

No.	付与されたタグ
1	[起動処理中][電源OFF/ON][動画撮影][特定フレームレート]
2	[連写][ファイル書出し処理中][検索再生]
3	[連写][ファイル書出し処理中][日付画面][日付跨ぎ]
4	[連写][ファイル書出し処理中][検索再生][電源OFF/ON]

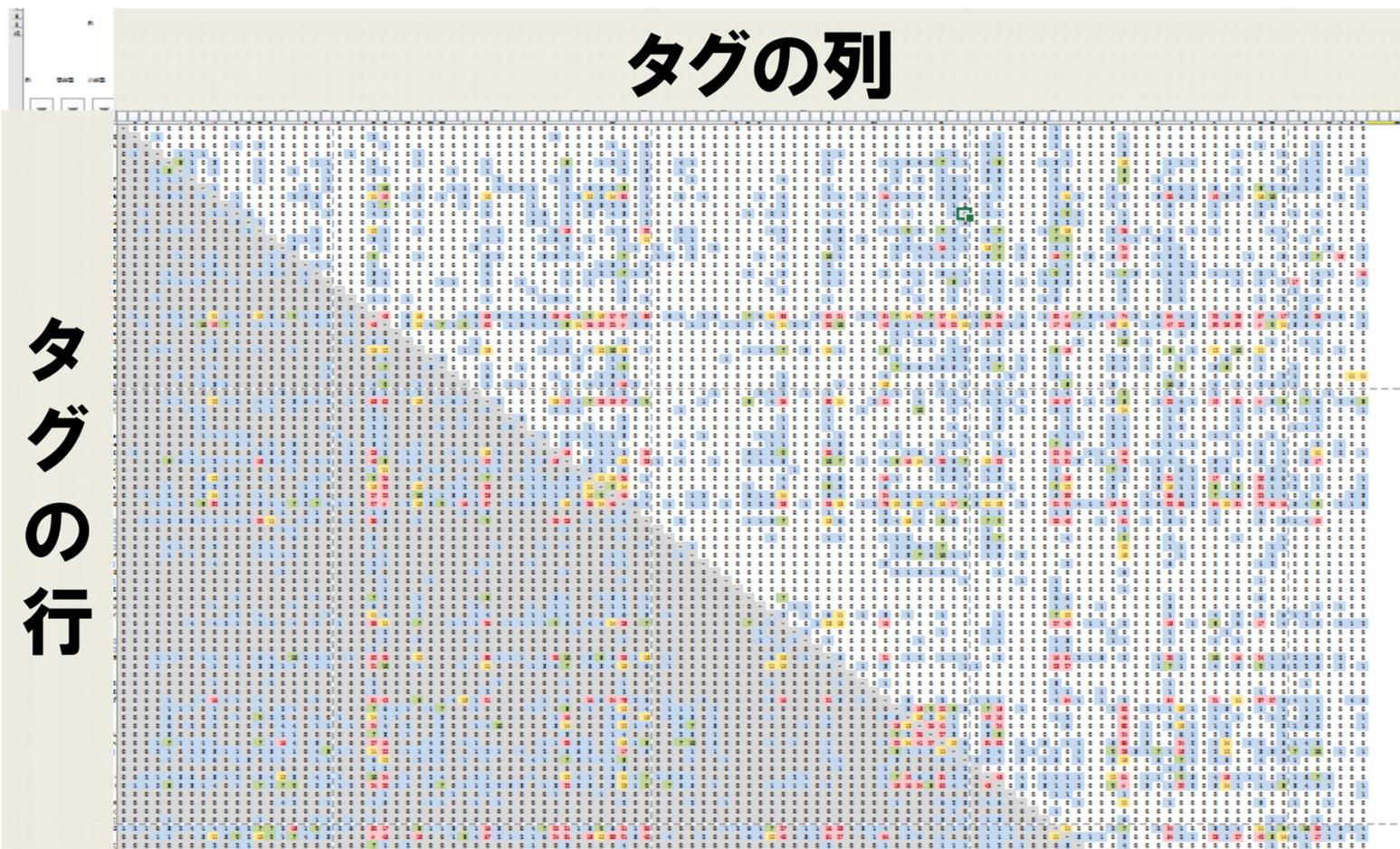
★前ページのタグをマトリクス化

	起動処理中	電源OFF/ON	動画撮影	特定フレームレート	連写	ファイル書出し処理中	検索再生	日付画面	日付跨ぎ
起動処理中		1	1	1					
電源OFF/ON			1	1	1	1	1		
動画撮影				1					
特定フレームレート									
連写						3	2	1	1
ファイル書出し処理中							2	1	1
検索再生									
日付画面									1
日付跨ぎ									

最後まで集計する

テスト設計とテスト密度の最適化

- リスクが高いエリア(赤セル)をピックアップしてテストを設計

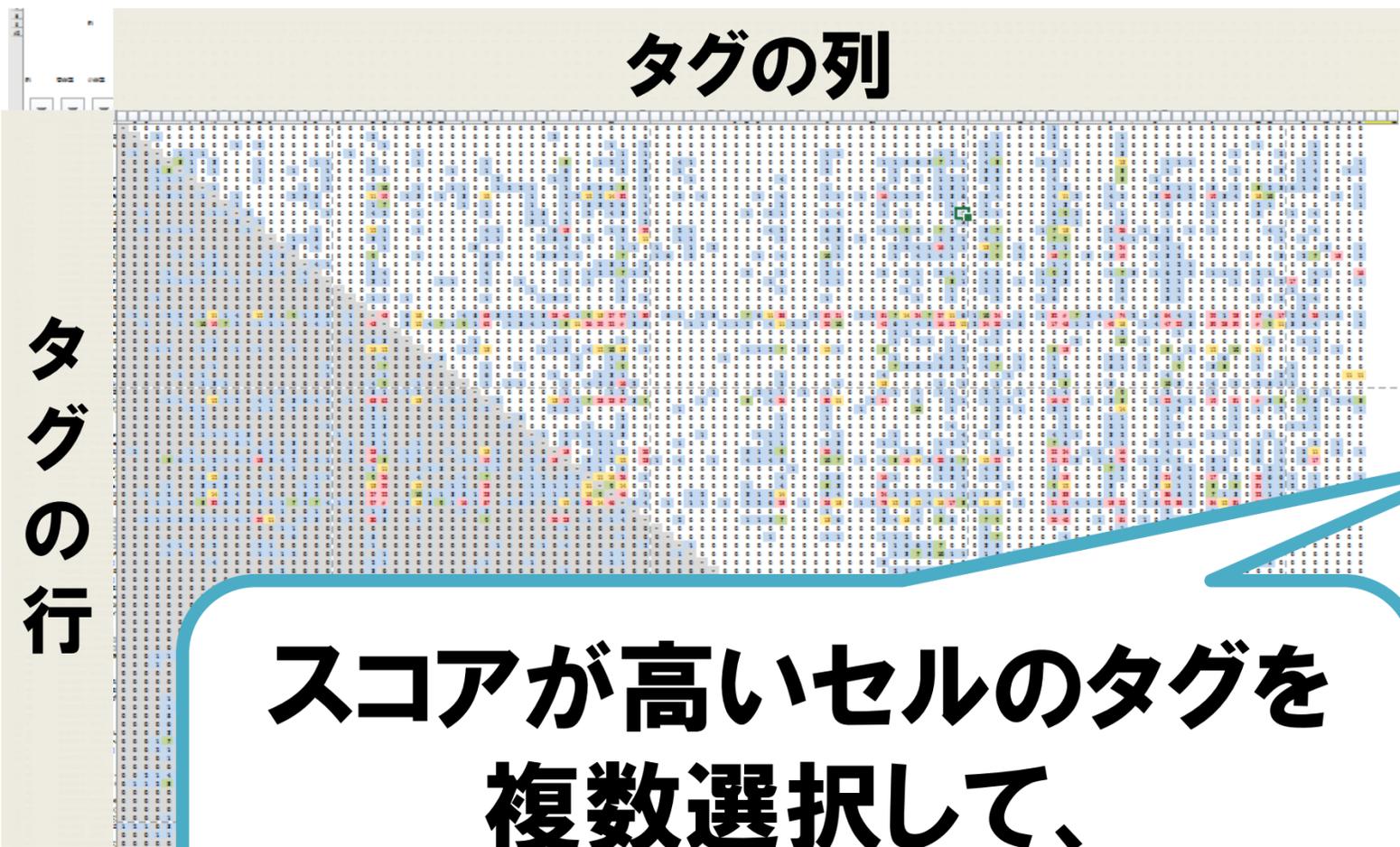


- テストケース書式
 1. 選定したタグ
 2. バグの狙いどころ
 3. 実施手順

(*) バグレポートにおけるバグタイトルの粒度の情報。
 例) 製品の設定をAに変更してから、機能Bを使用すると、機能Cが勝手に動作してしまう。 など

テスト設計とテスト密度の最適化

- リスクが高いエリア(赤セル)をピックアップしてテストを設計



スコアが高いセルのタグを複数選択して、狙いどころを検討する。

- テストケース書式
 1. 選定したタグ
 2. バグの狙いどころ
 3. 実施手順

(*) バグレポートにおけるバグタイトルの粒度の情報。
 例) 製品の設定をAに変更してから、機能Bを使用すると、機能Cが勝手に動作してしまう。 など

テスト設計とテスト密度の最適化

- バグへのタグ付与と同じ手法で、テスト状況も把握する
- テストケースにタグを付与して、タグマトリクスを作成する

バグが検出されている組み合わせを集計

秘 | CONFIDENTIAL

■ データ集計方法

- 以下の例の場合

No.	付与されたタグ
1	[起動処理中][電源OFF/ON][動画撮影][特定フレームレート]
2	[連写][書出し処理中][インデックス再生]
3	[連写][書出し処理中][日付ビュー][日付跨ぎ]
4	[連写][書出し処理中][インデックス再生][電源OFF/ON]

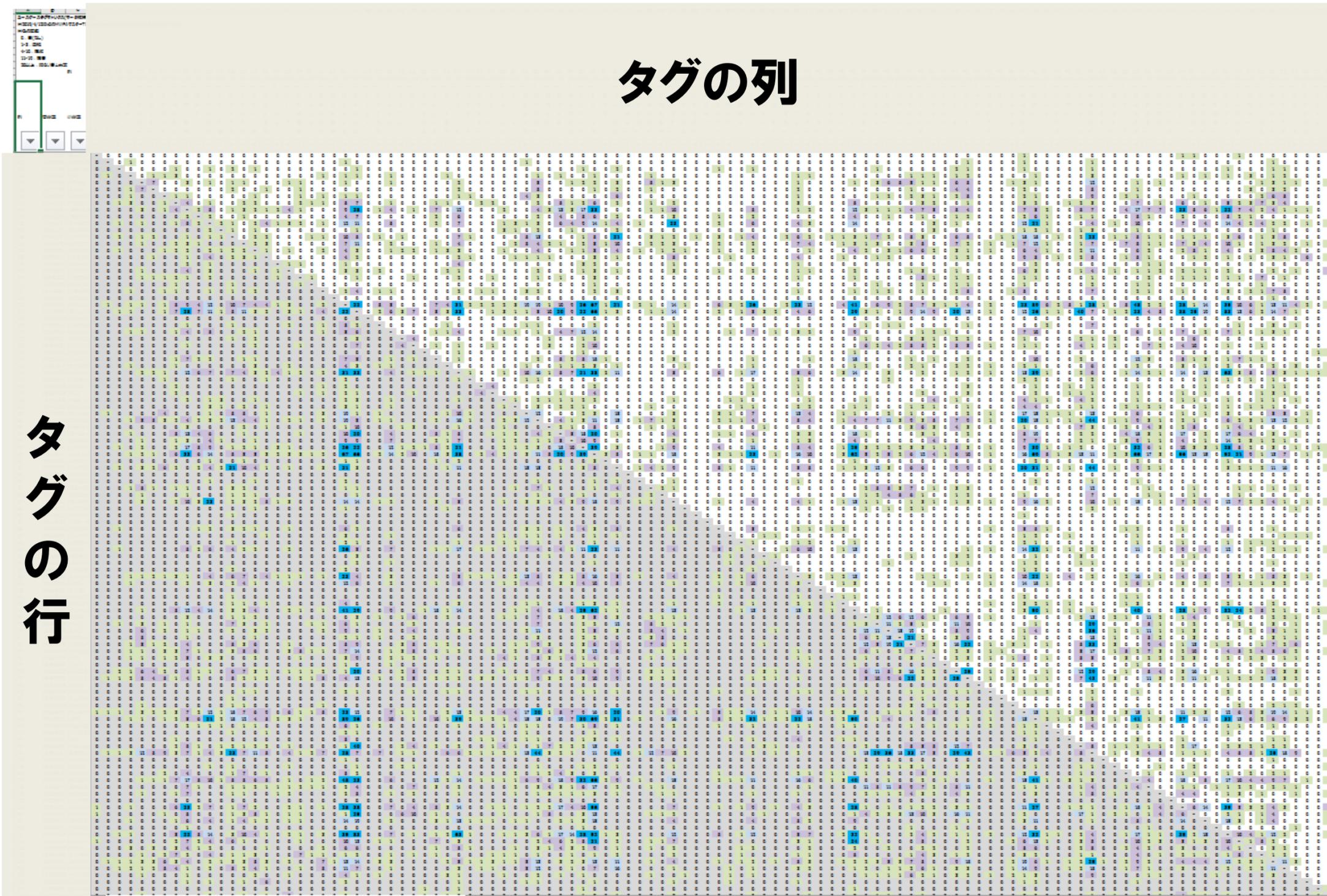
★前頁のタグをマトリクス化

	起動処理中	電源OFF/ON	動画撮影	特定フレームレート	連写	書出し処理中	インデックス再生	日付ビュー	日付跨ぎ
起動処理中		1	1	1					
電源OFF/ON			1	1	1	1	1		
動画撮影				1					
特定フレームレート									
連写						3	2	1	1
書出し処理中							2	1	1
インデックス再生									
日付ビュー									1
日付跨ぎ									

集計する

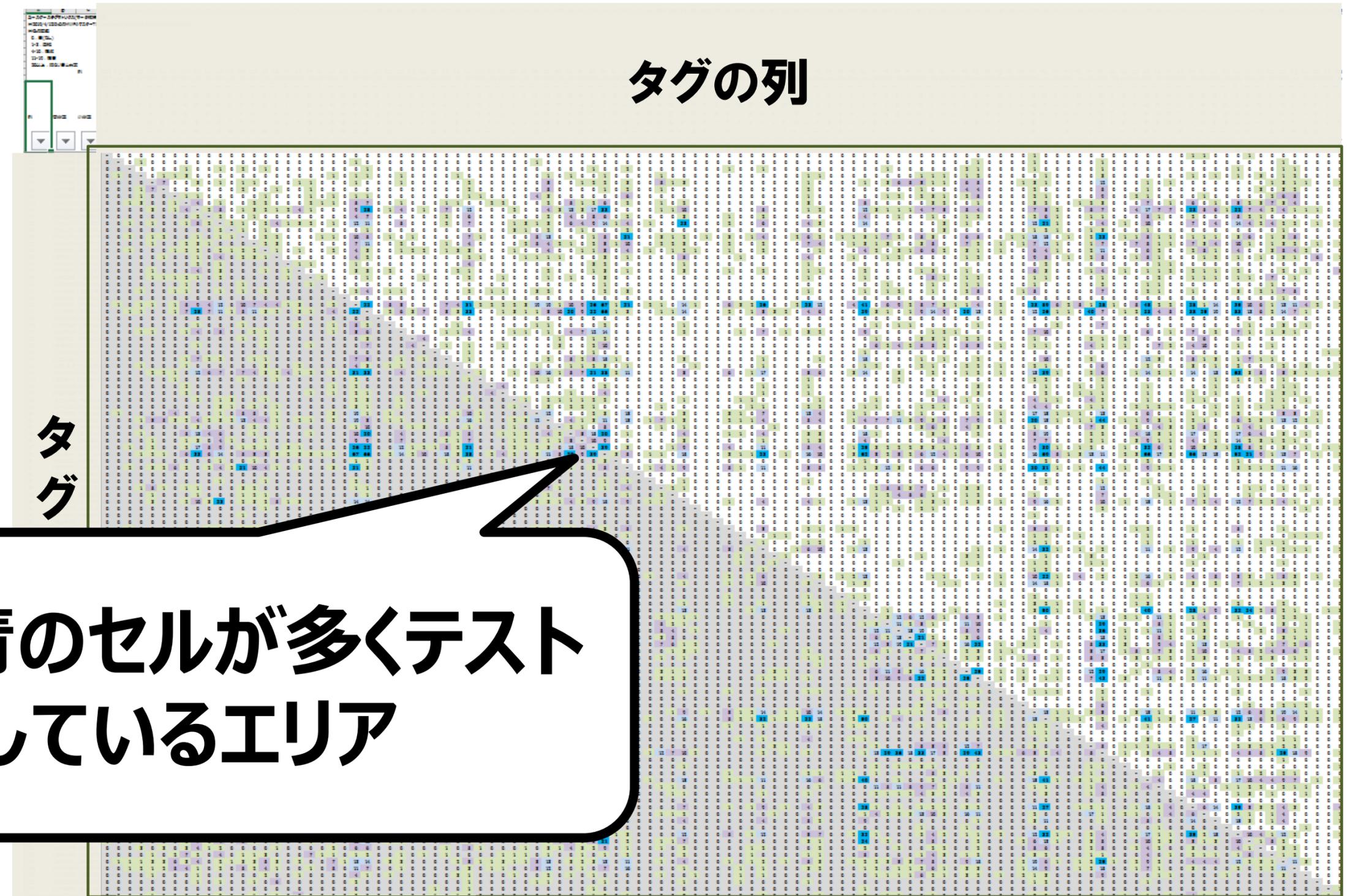
テスト設計とテスト密度の最適化

■ テストケースのタグマトリクス参考例(実物)



テスト設計とテスト密度の最適化

■ テストケースのタグマトリクス参考例(実物)

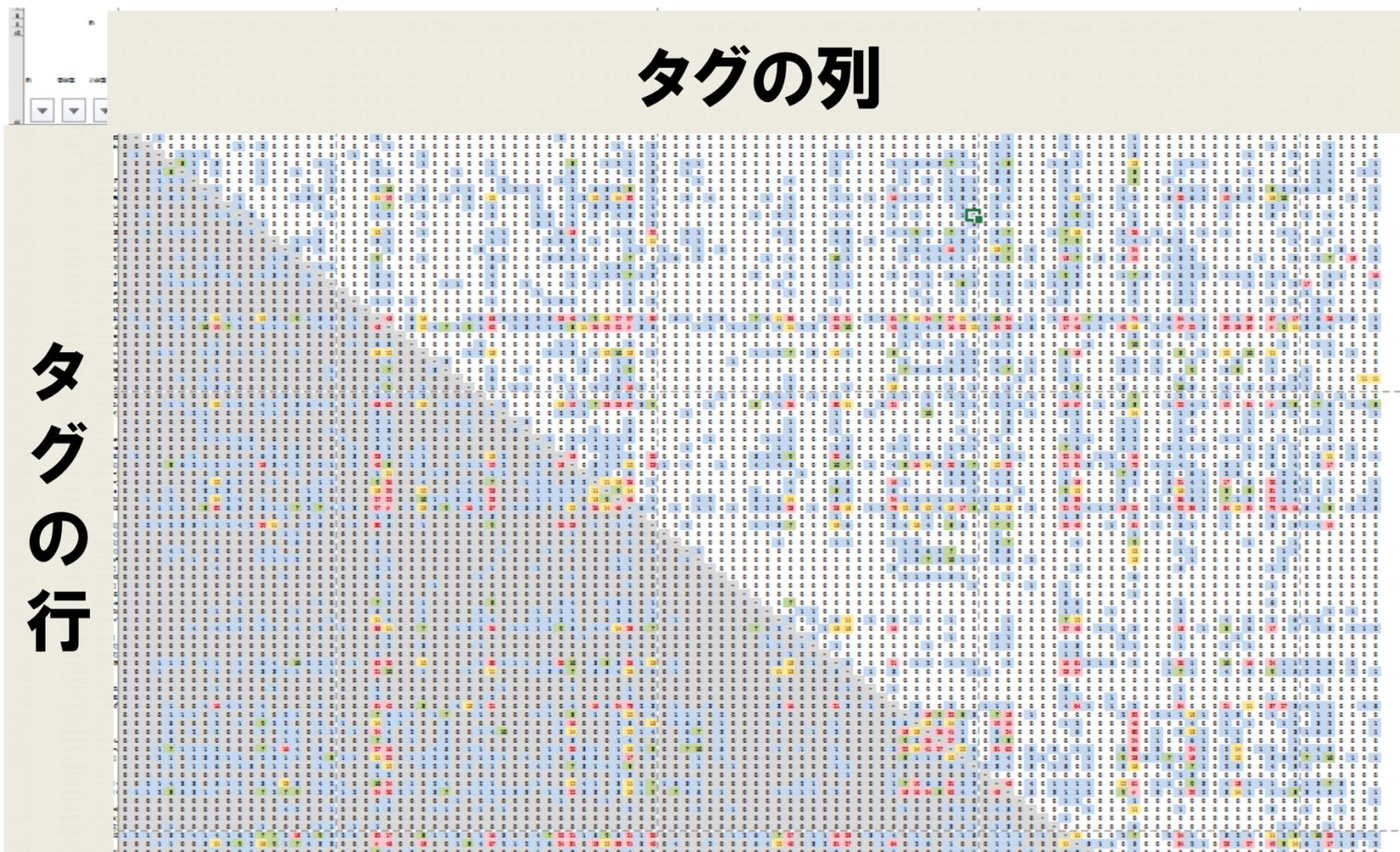


- 【凡例】
- 白：評価実施無し
 - 黄緑：評価1件以上
 - 薄紫：評価4件以上
 - 薄青：評価11件以上
 - 濃青：評価20件以上

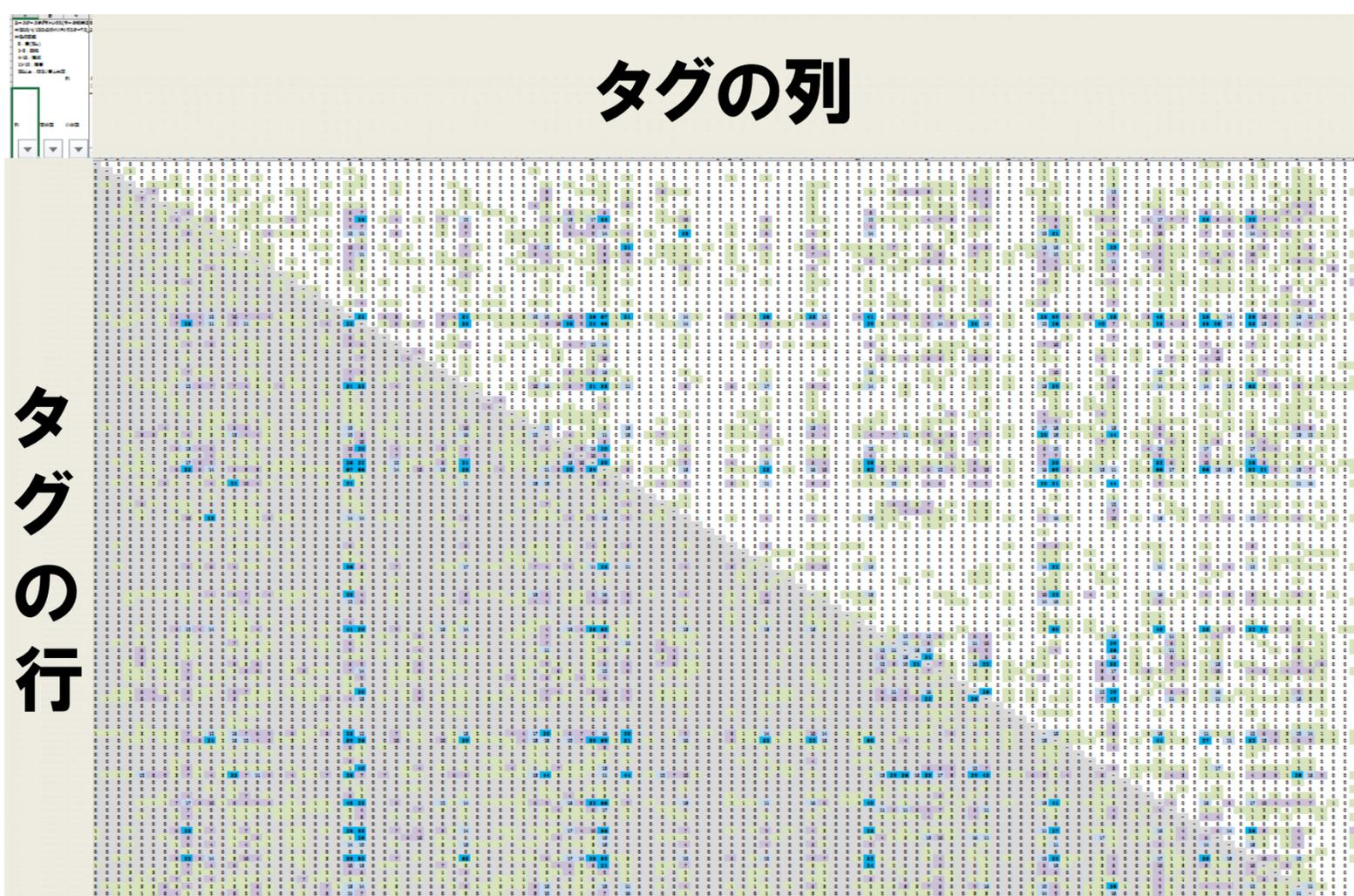
テスト設計とテスト密度の最適化

- バグ発生分布とテストケースの分布を照合して、テストケース数に対するバグ発生状況を把握する
 - 「バグ発生数」／「テストケース実施数」=『テスト密度』

●バグ発生の分布



●テストケースの分布

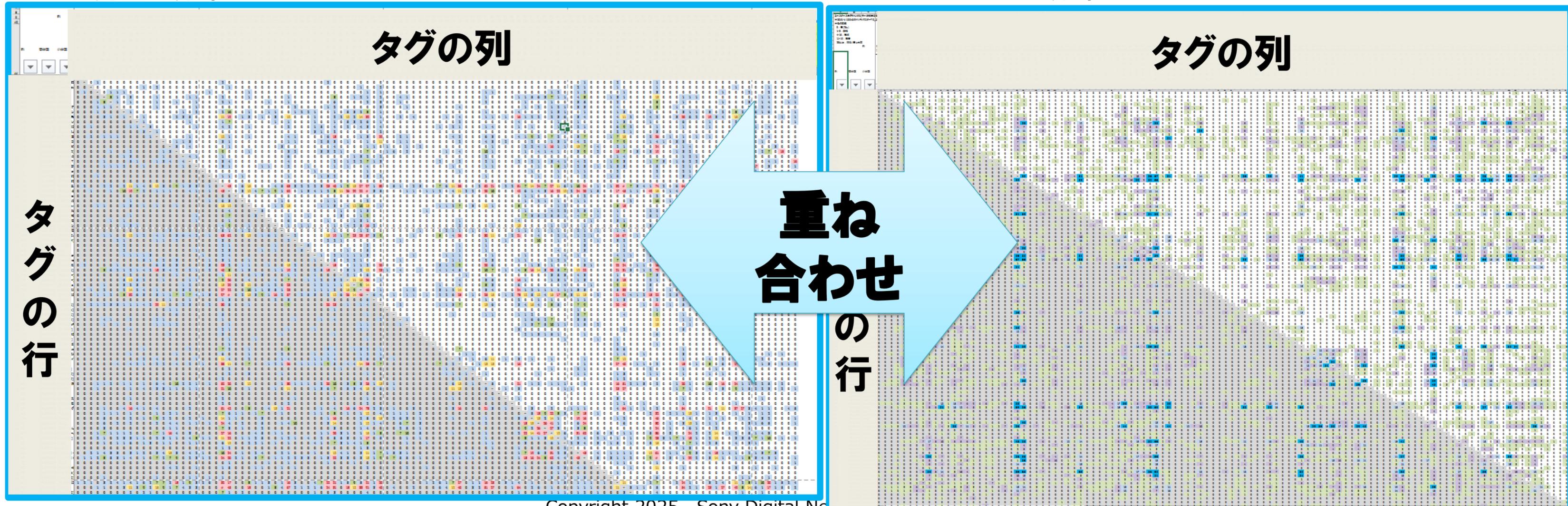


テスト設計とテスト密度の最適化

- バグ発生分布とテストケース分布を照合して、テストケース数に対するバグ発生状況を把握する
 - 「バグ発生数」／「テストケース実施数」=『テスト密度』

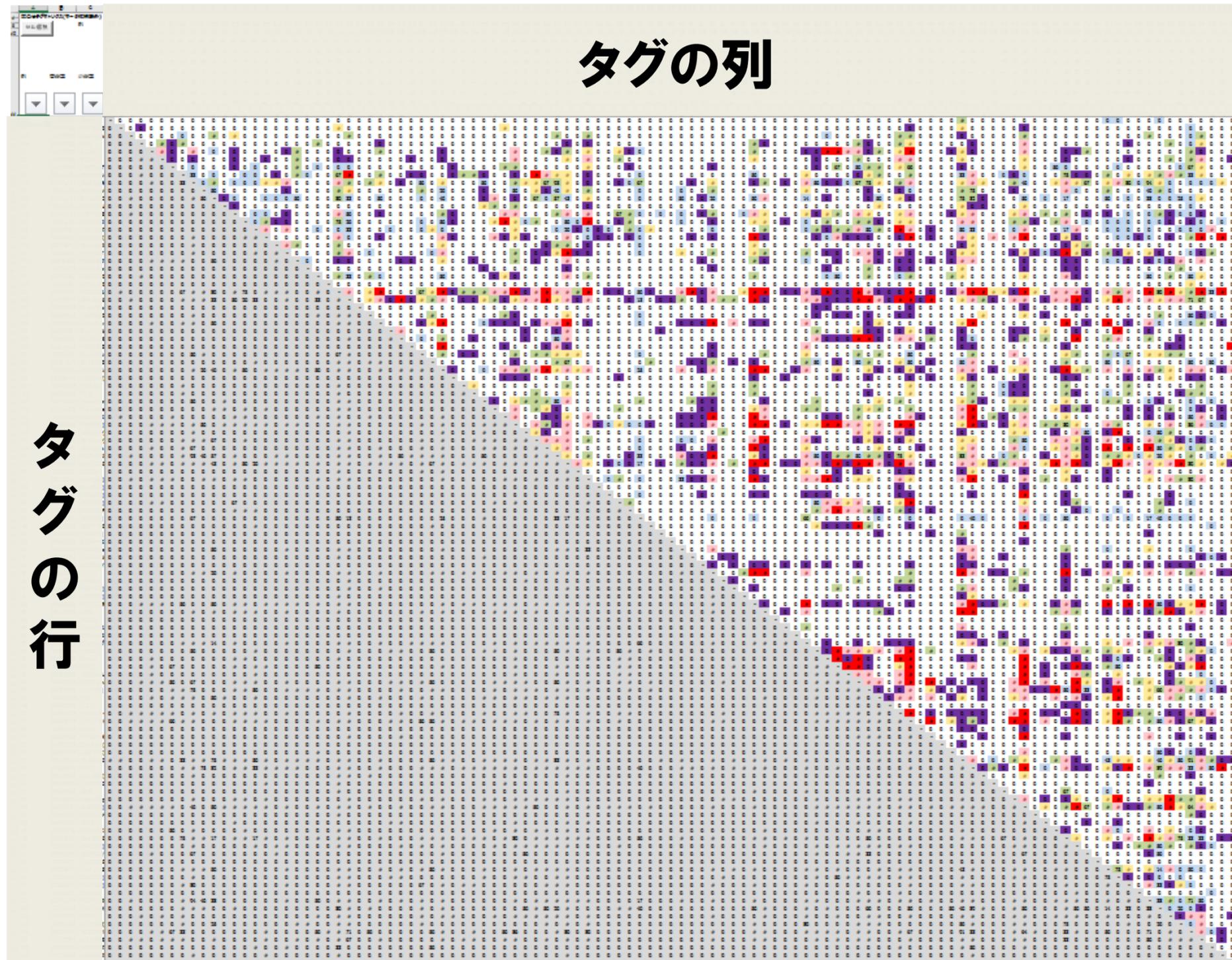
●バグ発生の分布

●テストケースの分布



テスト設計とテスト密度の最適化

■ タグマトリクス参考例(実物)：テスト実施に対するバグ発生状況



【凡例】

白：バグ発生&テストケース無し

青：バグ発生率0%以上

緑：バグ発生率50%以上

黄：バグ発生率100%以上

赤：バグ発生率200%以上

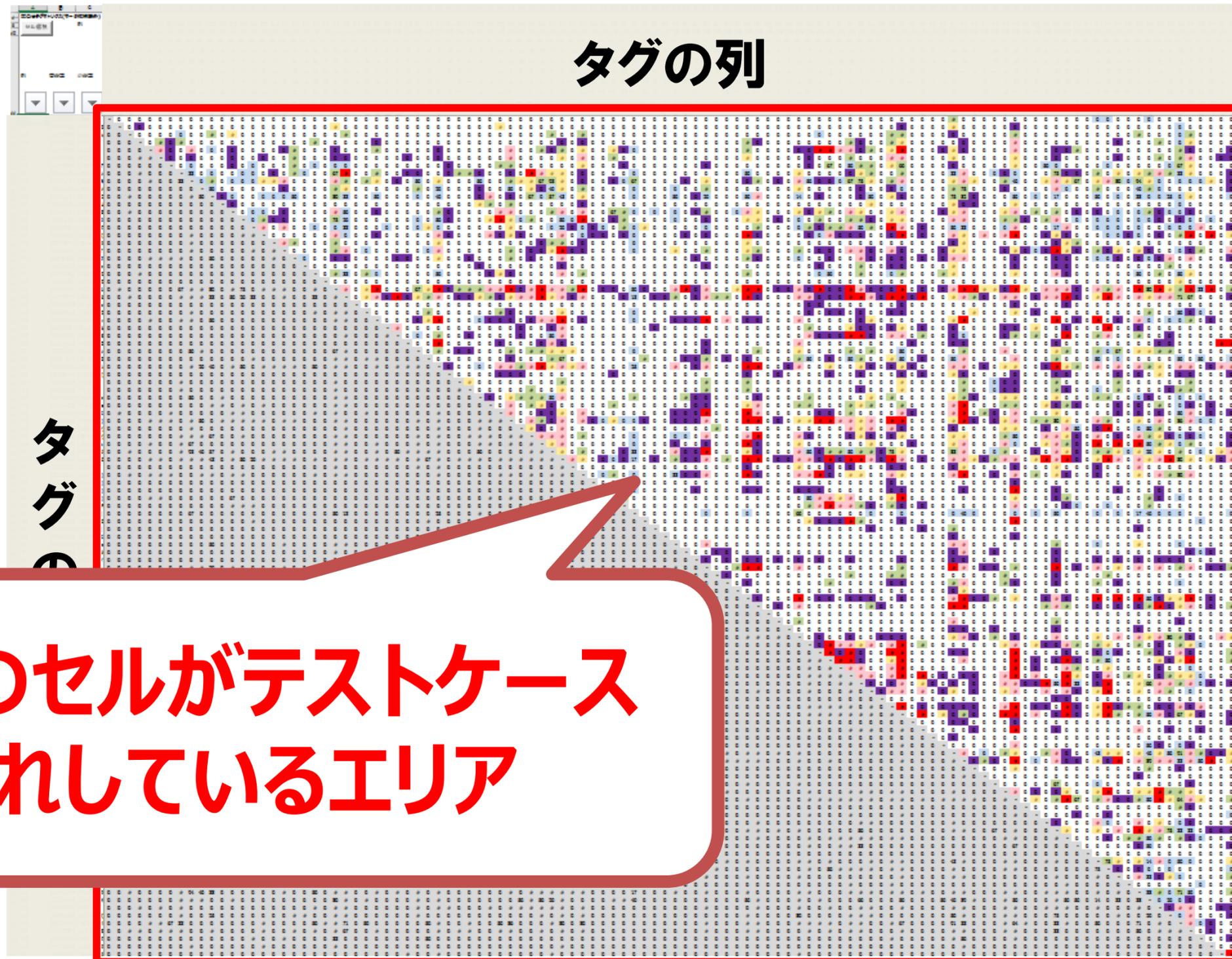
紫：バグ発生有り&テストケース無し

※1:バグ発生率=バグ発生数/テストケース

※2:評価実施有りでバグ0件は青に含む

テスト設計とテスト密度の最適化

■ タグマトリクス参考例(実物)：テスト実施に対するバグ発生状況



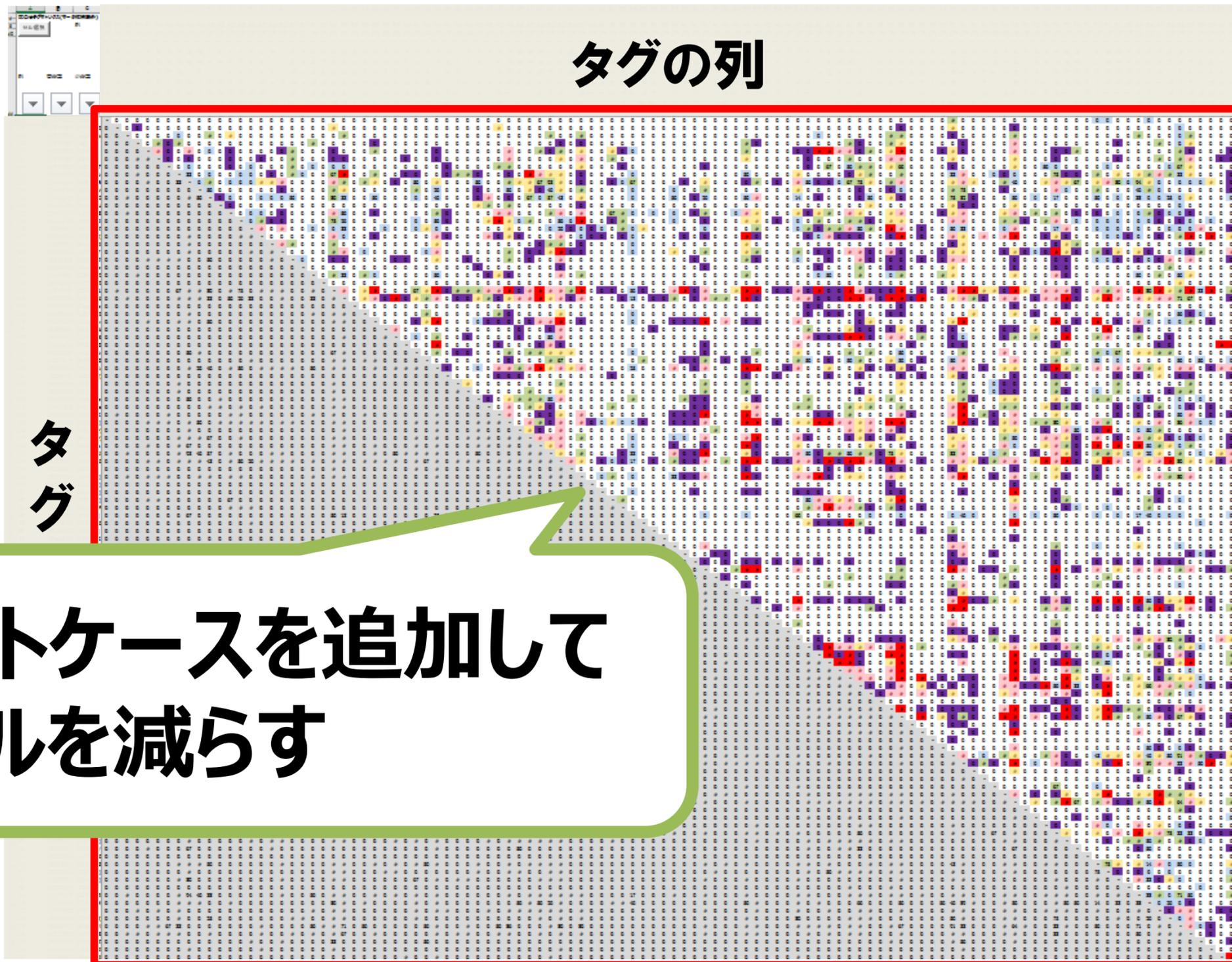
【凡例】

- 白：バグ発生&テストケース無し
 - 青：バグ発生率0%以上
 - 緑：バグ発生率50%以上
 - 黄：バグ発生率100%以上
 - 赤：バグ発生率200%以上
 - 紫：バグ発生有り&テストケース無し
- ※1:バグ発生率=バグ発生数/テストケース
 ※2:評価実施有りでバグ0件は青に含む

・紫色のセルがテストケース作成漏れしているエリア

テスト設計とテスト密度の最適化

■ タグマトリクス参考例(実物)：テスト実施に対するバグ発生状況



【凡例】

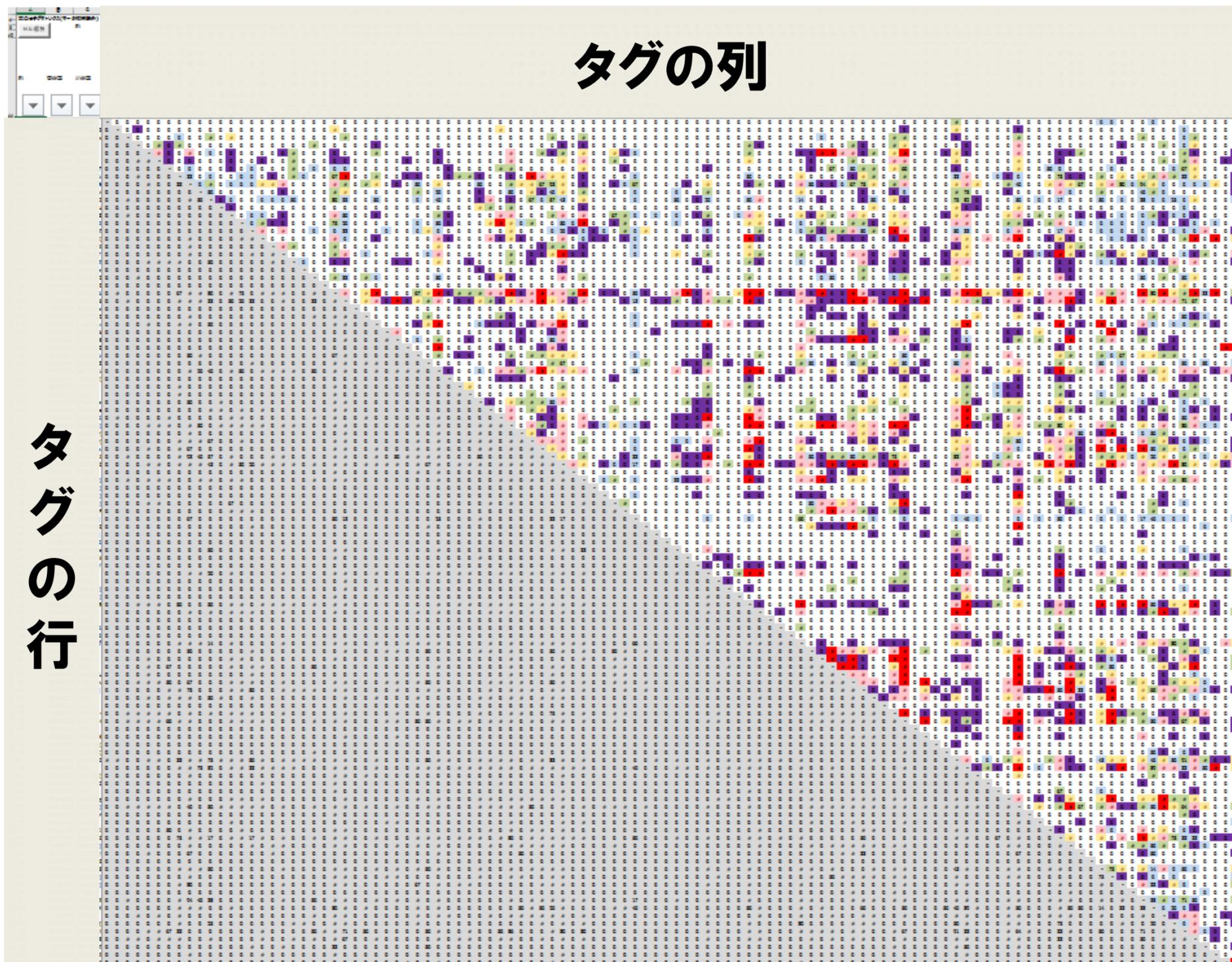
- 白：バグ発生 & テストケース無し
 - 青：バグ発生率0%以上
 - 緑：バグ発生率50%以上
 - 黄：バグ発生率100%以上
 - 赤：バグ発生率200%以上
 - 紫：バグ発生有り & テストケース無し
- ※1:バグ発生率=バグ発生数/テストケース
 ※2:評価実施有りでバグ0件は青に含む

⇒テストケースを追加して
紫セルを減らす

テスト設計とテスト密度の最適化

適正化前

■ タグマトリクス参考例(実物)：テスト実施に対するバグ発生状況



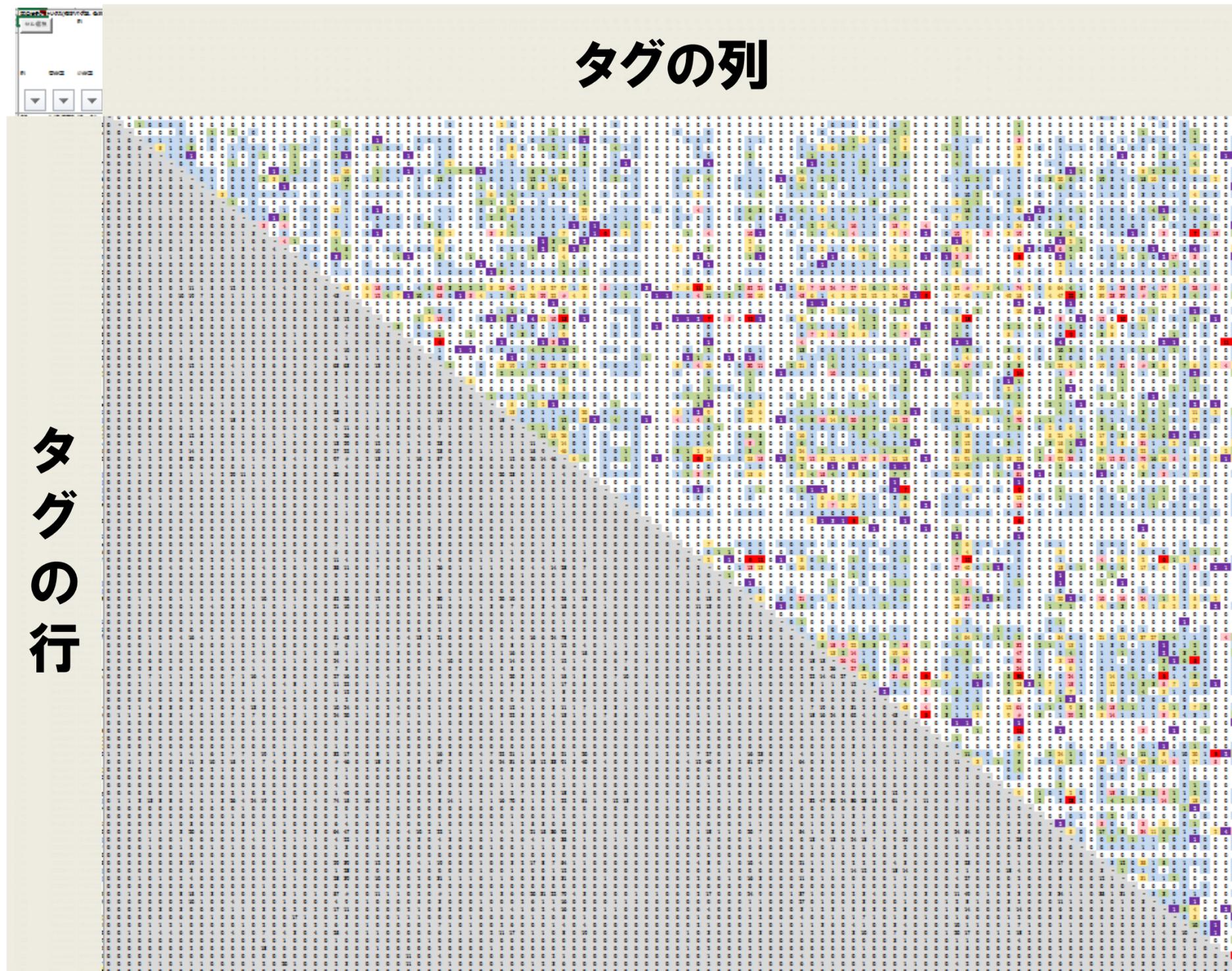
【凡例】

- 白：バグ発生 & テストケース無し
 - 青：バグ発生率0%以上
 - 緑：バグ発生率50%以上
 - 黄：バグ発生率100%以上
 - 赤：バグ発生率200%以上
 - 紫：バグ発生有り & テストケース無し
- ※1:バグ発生率=バグ発生数/テストケース
 ※2:評価実施有りでバグ0件は青に含む

テスト設計とテスト密度の最適化

適正化前

■ タグマトリクス参考例(実物)：テスト実施に対するバグ発生状況

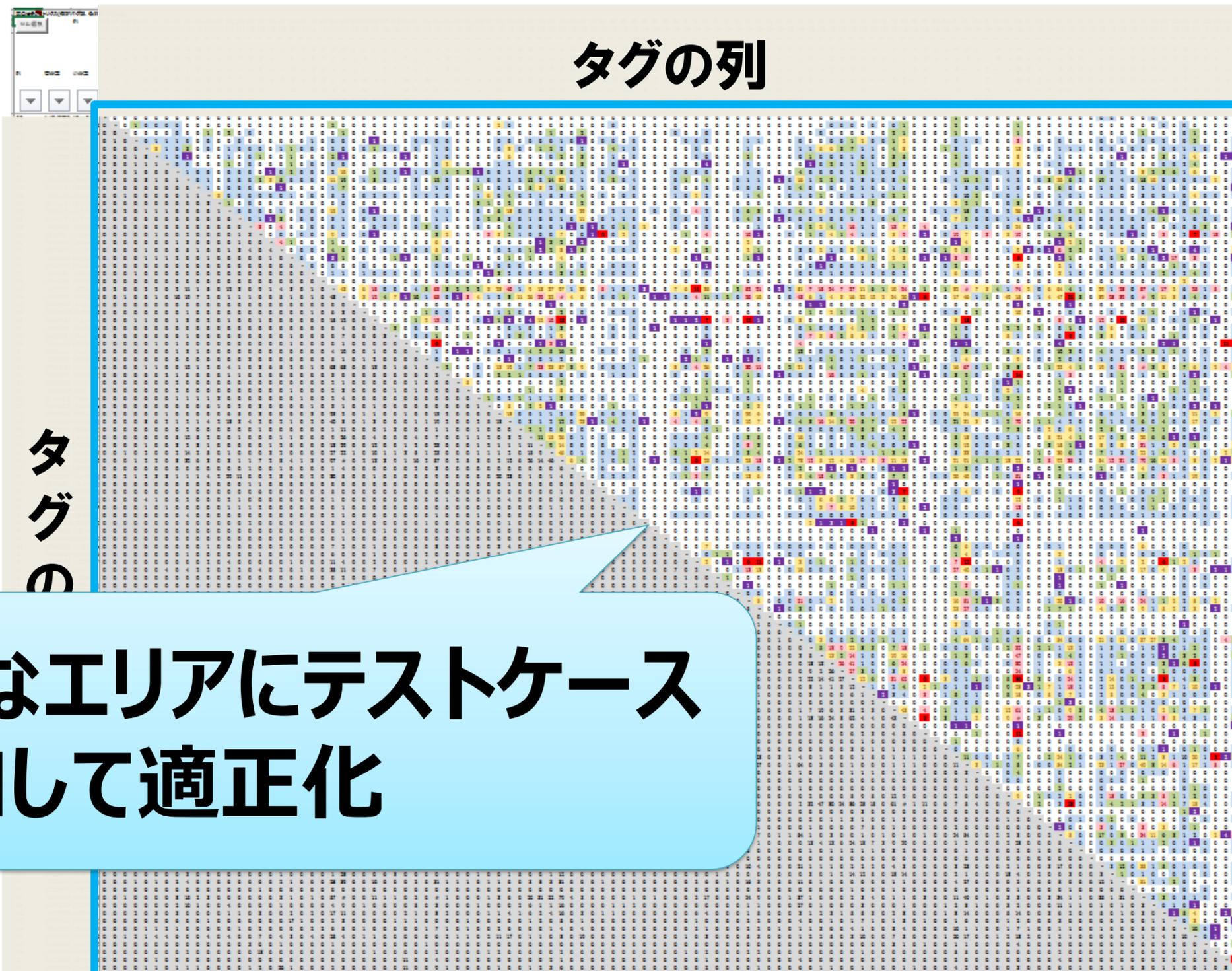


【凡例】

- 白：バグ発生&テストケース無し
 - 青：バグ発生率0%以上
 - 緑：バグ発生率50%以上
 - 黄：バグ発生率100%以上
 - 赤：バグ発生率200%以上
 - 紫：バグ発生有り&テストケース無し
- ※1:バグ発生率=バグ発生数/テストケース
 ※2:評価実施有りてバグ0件は青に含む

テスト設計とテスト密度の最適化

■ タグマトリクス参考例(実物)：テスト実施に対するバグ発生状況



【凡例】

- 白：バグ発生 & テストケース無し
 - 青：バグ発生率0%以上
 - 緑：バグ発生率50%以上
 - 黄：バグ発生率100%以上
 - 赤：バグ発生率200%以上
 - 紫：バグ発生有り & テストケース無し
- ※1:バグ発生率=バグ発生数/テストケース
 ※2:評価実施有りでバグ0件は青に含む

◎ 必要なエリアにテストケースを追加して適正化

タグマトリクスによる手法の横展開

➤ 「PMGテスト」 ⇒ 「Networkテスト」 ⇒ 「UI面テスト」
の順番に導入

テスト対象	特徴など	バグ検出率(導入年度の数値)
PMG	<ul style="list-style-type: none"> ・MW同様、発生するバグに特定の傾向が有る。 ⇒大きな問題も無く、比較的容易に導入できた。 	0.030(約3倍) ~ 0.098(約10倍)
Network	<ul style="list-style-type: none"> ・MW同様、発生するバグに特定の傾向が有る。 ⇒当初は有識者不在でバグレポートのみでタグを作成。しかし、有識者からの情報が必要という事になりタグの追加/見直しを行ってから導入となった。 	0.025(約3倍) ~ 0.093(約9倍)
UI面	<ul style="list-style-type: none"> ・発生するバグはPJによって異なり特定の傾向は無い。 ⇒『タグの作成基準』を定義してから、タグ作成に着手。UI面の一番広い範囲をカバーできるようにして、PJ毎の異なる傾向にも対応できるようにして導入した。 	0.018(約2倍) ~ 0.103(約10倍)

EOF